[redacted]

Technical Report No. 141

SACLANT ASW
RESEARCH CENTRE

SOUND FIELD COMPUTATION BY MEANS OF DESK TOP COMPUTERS

by

AUDUN SKRETTING

15 MARCH 1969

[redacted]

VIALE SAN BARTOLOMEO, 400
I-19026-LA SPEZIA, ITALY

[redacted]

[redacted]

AD085 2430

[redacted]

TECHNICAL REPORT NO. 141

SACLANT ASW RESEARCH CENTRE
Viale San Bartolomeo 400
I 19026 - La Spezia, Italy

SOUND FIELD COMPUTATION BY MEANS OF DESK TOP COMPUTERS

By

Audun Skretting

15 March 1969

APPROVED FOR DISTRIBUTION

Ir M.W. van Batenburg
Director

[redacted]

[redacted]

Manuscript Completed:
30 October 1968

[redacted]

TABLE OF CONTENTS

List of Figures

TABLE OF CONTENTS (Cont'd)

SOUND FIELD COMPUTATION BY MEANS OF DESK TOP COMPUTERS

By

Audun Skretting

ABSTRACT

Several formulae concerning sound in sea water are presented in forms that are convenient to program on a desk-top computer (DTC).

They are:

a) Formula for calculating sound speed in sea water.

b) Ray tracing formulae for calculating the sound ray path in a medium with the sound speed varying only as a function of depth.

c) Intensity formulae for calculating the intensity along a ray already plotted.

d) Travel-time formulae for calculating the travel-time for a ray already plotted.

e) Pressure-depth formula.

Special applications to one existing DTC (the Olivetti 101) are given, with the programs given in tables. Two examples to illustrate the program's usefulness are given at the end.

INTRODUCTION

The computation of sound fields in underwater acoustics started mainly during World War II [Ref. 1].

From then to the late 1950's, when ray-tracing programs for digital computers became more and more common, many different approaches were made to the problem [Refs. 2 and 3]. The numerical solution is not difficult, but to give a picture of the sound field requires many calculations to be made with far greater accuracy than is possible with a slide rule. The solutions were usually obtained with mechanical calculators, analogue computers, or numerical calculations by means of precomputed tables.

As access to big computers became common, these took over the task of ray-tracing.

During the last years, different types of desk-top computers (DTC) have been put on the market. They are digital "mini-computers", and despite their low price they have memory for both instructions and figures. This paper demonstrates how these can be used to solve ray-tracing problems.

The programs presented here for desk-top computers are not intended to compete with ray-tracing on large computers. But when a large computer is not available (on a research vessel, say, or in a small factory) or when it is overloaded, they are quite helpful for small ray-tracing problems. Also a short computation of the sound field by a DTC can be helpful when specifying the data for a large computer's ray-tracing program.

The report is divided into three parts.

Chapter 1 presents formulae for computing the sound field, with the simplifications necessary for a DTC.

Chapter 2 contains the application of the formulae for an existing DTC (the Olivetti 101).

Chapter 3 presents two examples of ray-tracing, illustrating the use of the different programs.

The appendix contains instructions for use of the given programs on an Olivetti 101 DTC. It also contains a short program with full description of the program steps so as to provide an insight into the programming methods.

1. THE FORMULAE

1.1 Sound Speed Formula

Wilson's second formula is at present widely used to calculate the speed of sound in sea water. Its complexity, however, necessitates the availability of electronic computers, for which it was primarily established.

In Ref. 4, C.C. Leroy proposed two formulae: one that approaches Wilson's second formula, and one that approaches the data Wilson used when computing his second formula. Although the formulae proposed by Leroy are simple, they are fairly accurate and it is demonstrated that the latter formula fits Wilson's data better than Wilson's equation. The two formulae are rather similar, and here the second is dealt with.

The formula is presented in Table 1. It has been divided into

a) A minimum number of terms, V_o, forming the "simplified" formula, which is sufficient with a slightly reduced accuracy for depths of less than 1000 m at any latitude provided that $t < 25°C$ and $30‰ < s < 42‰$.

b) A depth, temperature, and latitude dependent expression, V_a, which is necessary to operate deeper than 1000 m. It extends the validity of the formula to depths of 7000 m and improves the accuracy to ± 0.1 m/s (for $t \leq 23°C$).

c) A temperature correction, V_b, for $t \geq 25°C$. This correction still leaves the results at lower temperature within 0.1 m/s of Wilson's formula

and permits the calculation of sound speed for temperatures up to 34° C.

d) A correction term, V_c, for very great depths, necessary only if $d > 7000$ m.

e) A corrective term, V_d, for low salinity to be applied only when $s < 30‰$.

It follows that the way to make programs for sound speed when using Leroy's formula is to make one program for the basic formula, and two small programs for the deep-ocean term and the low-salinity term.

1.2 Ray Tracing Formulae

Figure 1 shows a sound ray in a layer of thickness Δz with constant sound speed gradient.

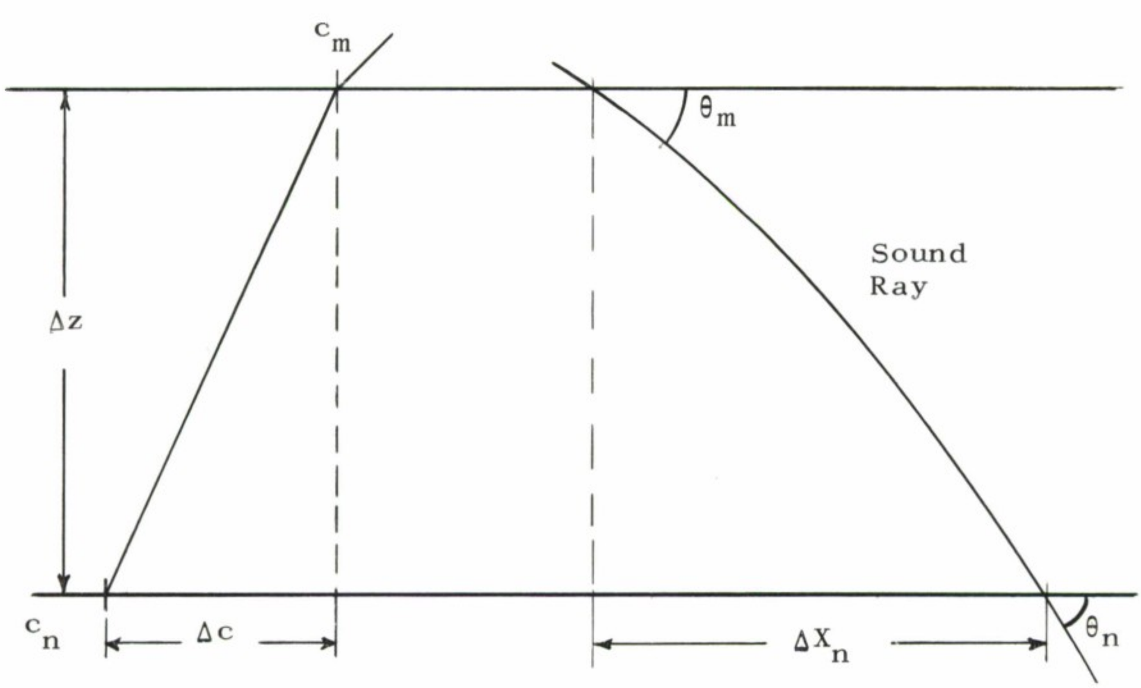

FIG. 1

TABLE 1

FORMULA FOR THE CALCULATION OF SOUND SPEED IN SEA WATER

COMPLETE $V = V_o + V_a + V_b + V_c + V_d$

BASIC $V = V_o + V_a + V_b$

SIMPLIFIED $V = V_o$

in which

$$V_o = 1493 + 3(T - 10) - 6 \times 10^{-3}(T - 10)^2 - 4 \times 10^{-2}(T - 18)^2 + 1.2(S - 35) - 10^{-2}(T - 18)(S - 35) + z/61$$

$$V_a = +10^{-1}\zeta^2 + 2 \times 10^{-4}\zeta^2(T - 18)^2 + 10^{-1}\zeta\Phi/90$$

$$V_b = 2.6 \times 10^{-4}t(t - 5)(t - 25)$$

$$V_c = -10^{-3}\zeta^2(\zeta - 4)(\zeta - 8)$$

$$V_d = +1.5 \times 10^{-3}(S - 35)^2(1 - \zeta) + 3 \times 10^{-6}t^2(t - 30)(S - 35)$$

Where V is the sound speed in m/s

T is the temperature in $^{\circ}$C

S is the salinity in ‰

z is the depth in m, and $\zeta = z/1000$, the depth in km

Φ is the latitude in degrees

Note: V_o can also be written:

$$V_o = 1449.44 + 4.56T - 0.046\,T^2 + 1.2(S - 35) - 10^{-2}(T - 18)(S - 35) + z/61$$

From Snell's law we have

$$\frac{c_n}{\cos \theta_n} = \frac{c_m}{\cos \theta_m} = k \, , \qquad \text{[Eq. 1]}$$

where k is a constant for the ray in any layer.

We introduce

$$\Delta c = c_m - c_n \, . \qquad \text{[Eq. 2]}$$

Ref. 5 gives the horizontal distance in one layer:

$$\Delta X = \frac{\Delta z \cdot k}{\Delta c} \cdot (\sin \theta_n - \sin \theta_m) \, . \qquad \text{[Eq. 3]}$$

As we have from Eq. 1 :

$$\cos \theta_n = \frac{c_n}{k} \, , \qquad \text{[Eq. 4]}$$

we get

$$\sin \theta_n = \sqrt{1 - \cos^2 \theta_n} = \sqrt{1 - \left(\frac{c_n}{k}\right)^2} \, . \qquad \text{[Eq. 5]}$$

By introducing Eq. 5 into Eq. 3 we have an expression for the horizontal distance for which it is possible to make a program. But if we examine Eq. 3, we notice that it contains two terms:

$$\frac{\Delta z \cdot k}{\Delta c} = r \, , \qquad \text{[Eq. 6]}$$

where r is the radius of the circle of which the sound ray forms an arc in the layer, and the term

$$\sin\theta_n - \sin\theta_m , \qquad \text{[Eq. 7]}$$

If we have a layer of nearly isovelocity water, Eq. 6 will have a very large value and Eq. 7 a very small value and the product will be rather unreliable because the computer must work with a restricted number of decimals. To alter the relative size of the two terms, and also to have a formula that is easy to evaluate, we use an approximation for $\sin\theta_n$.

From Eq. 5 we have

$$\sin\theta_n = \frac{1}{k}\sqrt{k^2 - c_n^2} = \frac{1}{k}\sqrt{(k+c_n)(k-c_n)} .$$

As $k + c_n \approx 2k$, we get

$$\sin\theta_n \approx \frac{1}{k}\sqrt{2k}\cdot\sqrt{k-c_n} = \sqrt{\frac{2}{k}}\cdot\sqrt{k-c_n} . \qquad \text{[Eq. 8]}$$

Equations 8 and 3 give

$$\Delta X = \frac{\Delta z}{\Delta c}\sqrt{2k}\cdot\left(\sqrt{k-c_n} - \sqrt{k-c_m}\right) . \qquad \text{[Eq. 9]}$$

By comparing Eq. 3 and Eq. 9 we see that the

$$r' = \frac{\Delta z}{\Delta c}\sqrt{2k} \qquad \text{[Eq. 10]}$$

of this formula is reduced about 27 times compared to r [Eq. 6]. This means that the term

$$\sqrt{k - c_n} - \sqrt{k - c_m} \qquad \text{[Eq. 11]}$$

is also increased 27 times compared to Eq. 7.

We still have the problem that if Δc goes to zero for one layer, Eq. 10 goes to infinity and Eq. 11 goes to zero. But this can easily be arranged in the following way.

The sound speeds (in m/s) put into the computer need be given to two decimal places only, because as the sound speed formula and the sound speed meter have a standard error of about 0.3 (m/s) there is no justification for taking more decimal places. The value of Δc must go in steps of 0.01. If $\Delta c = 0$, it can be made equal to 0.01 in the computer, and the error introduced is surely less than the error made by uncertainty in sound speed.

In Eq. 9, Δd must always be positive and Δc must always be equal to $c_m - c_n$ to obtain a positive value for ΔX.

Equation 9 is rather simple to compute. Some precautions must be taken however. If we give the computer a sound speed bigger than k, we may still get a result because some computers take the square root of the absolute value, thereby ignoring the fact that in this case we have a negative number and thus have no solution. But as $k = \frac{c_n}{\cos \theta_n}$ is constant for the ray, the ray is not able to penetrate into an area with sound speed greater than the k value. In fact the ray will turn when it reaches a sound speed equal to its k value, because

$$k = \frac{c_n}{\cos \theta_n} = \frac{c_n}{\cos 0} = c_n \quad \text{for} \quad \theta_n = 0° .$$

That means that in the program we must test whether the given c_n is bigger than the k value and, if so, compute the turning point and modify the input data.

The computation of the turning point is simple because, as for the rest of the computation, we suppose linear gradient.

The depth of the turning point (where $c = k$) is

$$z_{turn} = z_n - \frac{(c_n - k) \cdot (z_n - z_m)}{c_n - c_m} . \qquad [Eq. 12]$$

In the program we need the value of $k = \frac{c_n}{\cos \theta_n}$, n being any point on the ray. Normally k is either determined by the initial angle and sound speed, or by a turning point (where $k = c_t$). To compute k from the initial data a separate program must be made where c_o and θ_o are given as inputs.

Cosine θ can be computed by means of the series

$$\cos \theta = 1 - \frac{X^2}{2} + \frac{X^4}{4!} , \qquad [Eq. 13]$$

where X is the angle in radians. By using only three terms, one makes an error at $30°$ of about 3×10^{-5}, which corresponds to less than $0.1°$. As the ray-tracing program should not be used with angles bigger than about $30°$ because of the simplification in the formula, this accuracy for the cosine function should be sufficient.

1.3 Intensity Formulae

Reference 5 gives the formulae

$$\frac{I_o}{I_p} = - X \frac{\sin \theta_p}{\cos \theta_o} \frac{dX}{d\theta_o} , \qquad [Eq. 14]$$

which is valid regardless of the path followed by the ray between the starting point and the point of interest p.

To compute this equation, we need an expression for $\frac{dX}{d\theta_o}$. Assuming, as before, that we can divide the ocean into horizontal layers of constant sound-speed gradients, Ref. 5 gives the formula:

$$a_n = \left[\Delta\left(\frac{dX}{d\theta_o}\right)\right]_m^n = r_n \tan \theta_o \left(\frac{1}{\sin \theta_m} - \frac{1}{\sin \theta_n}\right) . \qquad [Eq. 15]$$

Here, r_n is the curvature radius in layer n, and a_n is the amount by which $\frac{dX}{d\theta_o}$ is changed by passing from layer m to layer n. Therefore:

$$\frac{dX_p}{d\theta_o} = \sum_{n=1}^{p} a_n . \qquad [Eq. 16]$$

By using the same approximation for $\sin \theta_n$ in Eq. 15 as before [see Eq. 8] we get:

$$a_n = \frac{\Delta z}{\Delta c} \sqrt{2k} \frac{k}{2} \tan \theta_o \left[\frac{\pm 1}{\sqrt{k - c_m}} - \frac{\pm 1}{\sqrt{k - c_n}}\right] . \qquad [Eq. 17]$$

The ray-tracing program must print out the value [Eq. 10]:

$$r'_n = \frac{\Delta z}{\Delta c} \sqrt{2k}$$

and it will be used as input data here.

Combining Eqs. 10, 14, 16 and 17 gives:

$$\frac{I_o}{I_p} = -X\left(\sin \theta_p\right)\frac{k}{2} \sum_{n=1}^{p} r_n{}' \frac{\tan \theta_o}{\cos \theta_o} \left[\frac{\pm 1}{\sqrt{k - c_m}} - \frac{\pm 1}{\sqrt{k - c_n}}\right]. \qquad \text{[Eq. 18]}$$

In special cases there are some problems in evaluating the formula; these are all discussed in Ref. 5.

The first problem is when $\theta_o = 0$. Then Eq. 6 cannot be used, and with the program made [see Ch. 2] it is not possible to calculate the intensity for a ray with an initial angle of zero.

The next problem is when the ray turns at point n so that $\theta_n = 0$. If we look at Eq. 15, we see that it is not possible to use this point. But we have $r_n = r_{n+1}$ and $\sin \theta_{n+1} = -\sin \theta_m$. Thus we can drop point n and go to n+1. Then $\sin \theta$ will change sign, and in the computation of Eq. 18 we must change the sign of the approximated $\sin \theta$ term. This can be done by storing a ± 1 to use in the denominator, and changing its sign every time a turning point is passed. By this simple approach, it is not possible to get the intensity in the turning point.

1.4 Travel-time Program

When a ray tracing is made, one often realizes that a certain point can be reached by several sound rays. To find the succession of the different arrivals, it is necessary to compute the travel time.

From Ref. 6 we have the formulae

$$t = \frac{X}{k} + \int \sqrt{\frac{1}{c^2} - \frac{1}{k^2}}\; dz \qquad \text{[Eq. 19]}$$

for the travel-time in seconds

$$X = \frac{1}{k} \int \frac{dz}{\sqrt{\frac{1}{c^2} - \frac{1}{k^2}}} \; . \qquad \text{[Eq. 20]}$$

Differentiating Eq. 20 gives

$$dX = \frac{1}{k} \frac{dz}{\sqrt{\frac{1}{c^2} - \frac{1}{k^2}}} \; . \qquad \text{[Eq. 21]}$$

Eliminating dz from Eq. 19 by means of Eq. 21 gives

$$t = \frac{X}{k} + k \int \left(\frac{1}{c^2} - \frac{1}{k^2}\right) dX \; , \qquad \text{[Eq. 22]}$$

$$t = \frac{X}{k} + k \int \frac{(k+c)(k-c)}{k^2 c^2}\, dX \approx \frac{X}{k} + \frac{2}{k^2} \int (k - c)\; dX \; . \qquad \text{[Eq. 23]}$$

Here we have made the approximation that $k + c \approx 2k$ and $k^2 c^2 \approx k^4$.

The integral $\int (k - c)dX$ is not easy to evaluate, since c is a linear function with depth in each layer, but is a complex function of the horizontal distance X. However, by selecting the layer thickness so that sound speed differences are not too great within each layer, we can use the approximation

$$c(a) \approx c_m + \frac{c_n - c_m}{\Delta X} \cdot a \ . \qquad [Eq. 24]$$

Equations 5 and 6 give

$$t \approx \frac{X_p}{k} + \frac{2\Delta X}{k^2} \sum_{n=1}^{p} \left(k - \frac{c_m + c_n}{2}\right) . \qquad [Eq. 25]$$

Equation 25 contains the k value, the sound speed of each layer, and the horizontal distance. It will normally be difficult to find space in a DTC to calculate the travel-time in the main ray-tracing program. But when the ray-tracing is done and the horizontal distance Xp is known, a small and easy program finds the travel-time.

2. APPLICATION OF THE FORMULAE

2.1 Introduction

There are already a number of different DTC's on the market and the number will surely increase in the future. To give recommendations for the programming of different sizes of DTC is outside the scope of this report. This section gives and explains the flowcharts that are linked to the special DTC for which the programs are made (Olivetti 101), but they are general and can serve as guides when programming on other DTC's. However, if the DTC being used has smaller memory and programming facilities, the programming of the presented formulae might be difficult or impossible.

If some details of the explanation are not clear, it might be helpful to study the Appendix, which gives a short description of the actual DTC, the programs in table form, and a description of the use of the programs. The Appendix also contains an example in which each step of the programming is explained.

The numbers indicated on the flowchart [Figs. 2, 3, 4 and 5] correspond to instruction numbers in the corresponding tables [Tables 9, 12, 13, 14 respectively].* The number of constants that can be put into a register and the number of constants it is necessary to generate in the computer will vary from computer to computer. The actual organization for the Olivetti 101 is therefore only mentioned in the Appendix.

The programs for computation of the soundfield are all based on a sound-speed distribution varying only as a function of depth. The first step is therefore to approximate the sound

---
* See programs between coloured pages at end of main text.

speed profile with straight segments, thus dividing the medium into layers of constant sound-speed gradient. If data are needed at a depth that does not coincide with a depth where the gradient is changing, it is necessary to compute the sound speed at this depth. This will often be the case for the source depth.

The programs work in series: one first uses the sound-speed program to get the sound speed profile and then uses the ray-tracing program. These provide input for the intensity program or the travel-time program.

2.2 Sound speed program [Fig. 2]

The formulae presented in Sect. 1.1 can easily be matched to different sizes of DTC. Here the formulae are programmed in two programs:

a) One program for the basic formula $V = Vo + V_a + V_b$.

b) One program contains the great-depth term Vc and the low salinity term Vd. They are programmed in series, so one calls the program for Vd or Vc by using different labels on the keyboard.

The program for the basic formula will cover most ocean cases. When the depth is greater than 7000 m or the salinity is less than 30‰, one has to use the other program to find the correction term.

The term Vo is rewritten to

$$Vo = 1401.14 + 49.1\ t/10 - 4.6\ (t/10)^2 + 13.8\ s/10 - (t/10)(s/10) + z/61.$$

Only the flowchart for the main program is shown in Fig. 2, the other program is so short and simple that a flowchart would be superfluous.

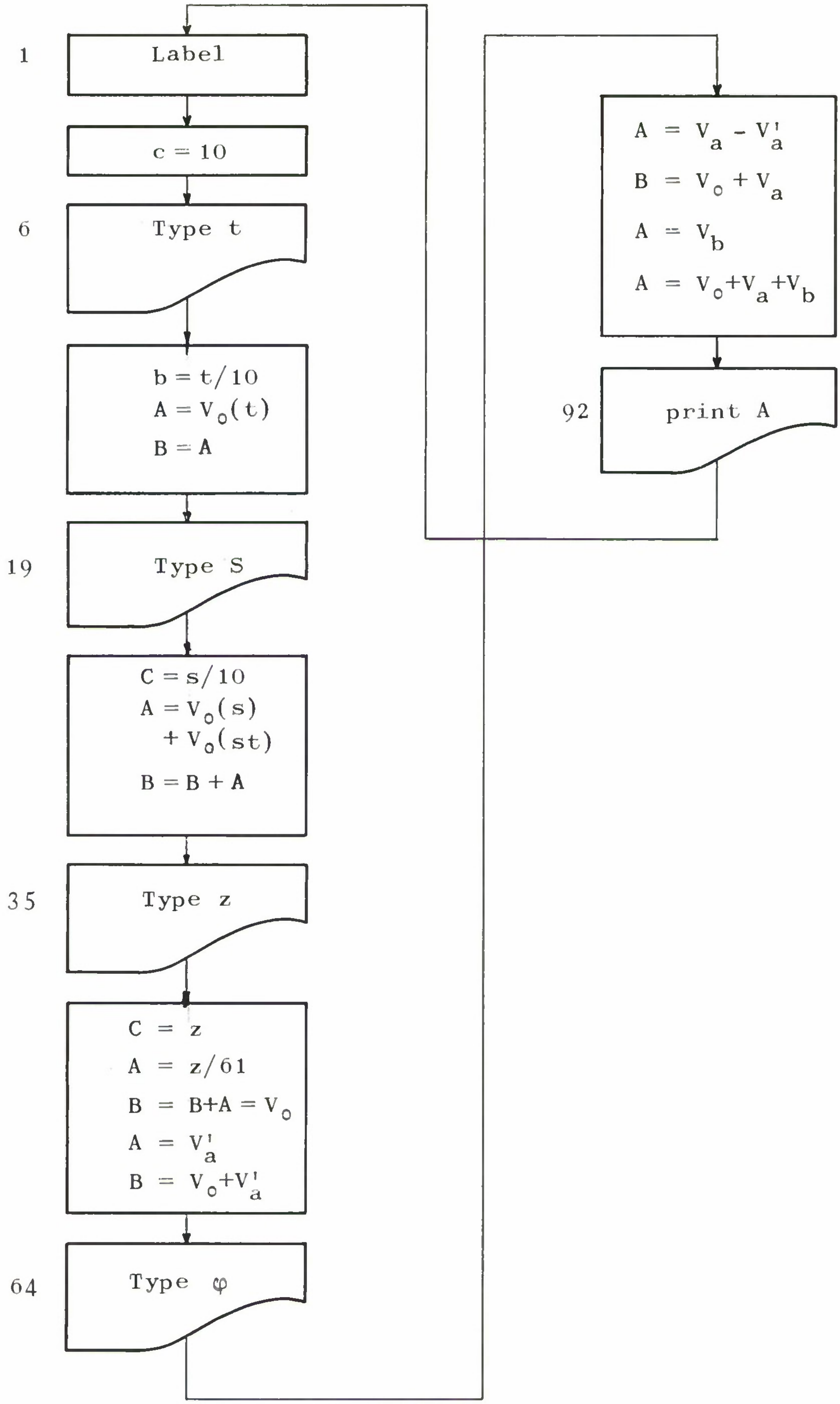

FIG. 2 FLOWCHART – SOUND SPEED PROGRAM

2.3 Ray tracing program [Fig. 3]

Because of space problems, one short program (not illustrated) is used to compute the k value:

$$k = \frac{c_0}{\cos \theta_0} .$$

For the DTC with $\cos \theta$ or $\sin \theta$ as a built-in subroutine, this is of course very simple. Here $\cos \theta$ is computed by using Eq. 13. The k value is then used as an input parameter to the ray tracing program.

For the ray tracing program Eqs. 9 and 12 are used. It must also be remembered that the program must print out $\tan \theta_0$ and r' [Eq.10] for the intensity program [Sect. 2.4]. As $\tan \theta$ had to be computed, the program was arranged so that this computation and printing is done for all θ_p.

The start of a program is reached by calling a label. The first part of the program receives the initial data, the k value, c_0 (sound speed at source), and z_0 (depth of source), and arranges them in registers. Take note of Instr. 8 which cleans Register D after new initial data; the sum of ΔX (horizontal range in each layer) is stored in this register. At the end of the initial data section, the computer is told to jump to the last part of the program for computing and printing $\tan \theta_0$.

The main loop of the program first receives the sound speed of the actual layer as input data. It stores the value, and computes and stores the Δc. In Instr. 18 the program checks if $\Delta c = 0$, and if so, the sound speed c_n and the Δc is modified by 0.01 m/s.

At Instr. 28, the depth of the layer (z_n) is given as input, stored, and the Δz_n computed.

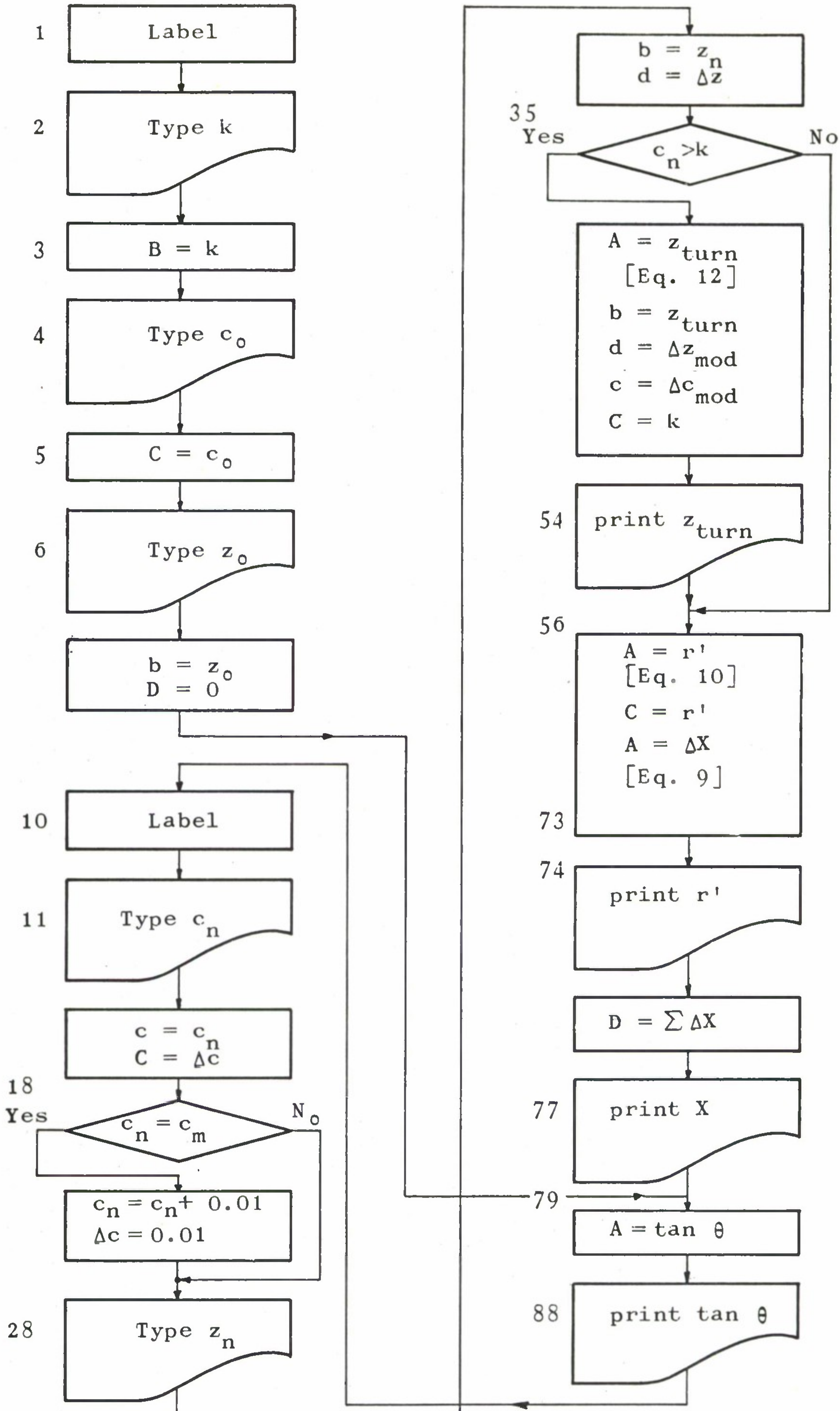

FIG. 3 FLOWCHART - RAY TRACING PROGRAM

Instr. 35 compares the sound speed with the k value. If the sound speed is smaller or equal to the k value, the computer jumps to Instr. 55. If the sound speed is greater than the k value, then we have asked for a sound speed and corresponding depth that the ray does not reach, because it turns beforehand. The computer then computes the turning point [Eq. 12], prints out the value, and corrects the values for c_n, Δc_n, z_n and Δz_n.

At Instr. 56 the evaluation of the formula starts. Instruction 74 prints out the value of r' [see Eq. 10], which is needed as input for the intensity program. As we have the same horizontal range whether Δz is positive or negative, we have to take the absolute value of the result. The program sums all the results in Register D and prints out the sum.

Instructions 79 to 90 compute tan θ_n and print out the value. This figure is not necessary, only tan θ_0 being used for the intensity program. By putting these instructions between Instrs. 8 and 10, the computer program will be a bit faster. But very often it will be convenient to know the angle in the various layers.

2.4 Intensity program [Fig. 4]

Equation 18 is used. The evaluation of the formula is made under the assumption that one is only interested in the intensity at certain points on the ray. Only the term

$$\alpha = r_n' \frac{\tan \theta_0}{\cos \theta_0} \left[\frac{\gamma}{\sqrt{k - c_m}} - \frac{\gamma}{\sqrt{k - c_n}} \right] \qquad \text{[Eq. 26]}$$

(where $\gamma = +1$ or -1)
is evaluated and summed for each set of input data.

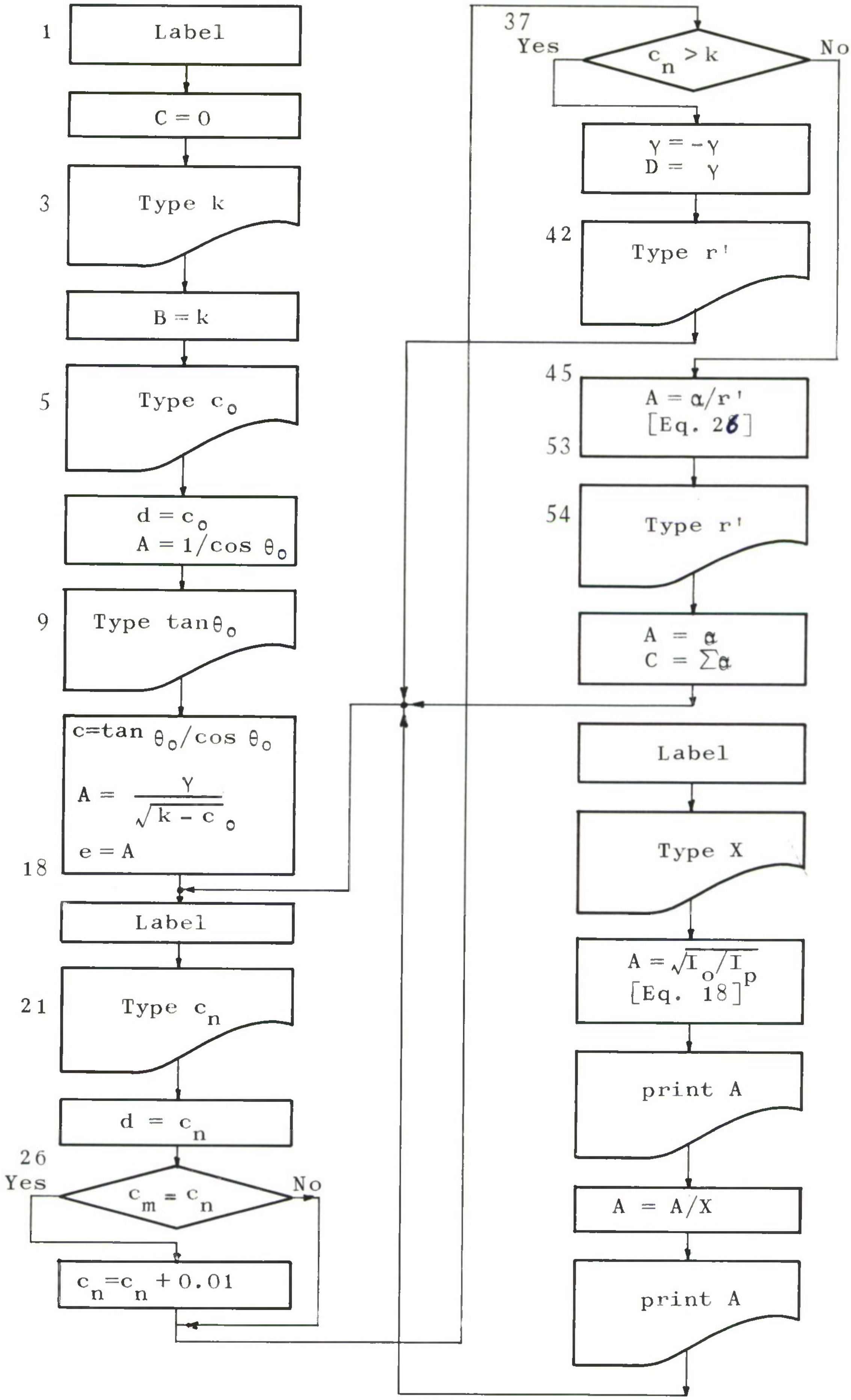

FIG. 4 FLOWCHART – INTENSITY PROGRAM

When the intensity at a point is wanted, one calls the last part of the program by pressing a label. Horizontal distance X is given as an input parameter, and the result is printed out.

For the final result, the product $\frac{k}{2} \sin \theta_p$ is needed. It is easy to find, by means of the same approximation for $\sin \theta$ as used before, that

$$\frac{k}{2} \sin \theta_p \approx \sqrt{\frac{k}{2}} \quad \sqrt{k - c_p} \quad . \qquad \text{[Eq. 27]}$$

Since I_o/I_p can often be a large number, care must be taken not to get overflow in the registers. Large numbers are also difficult to read, e.g. a not unusual loss of 75 dB will be printed out as 31620000. In the program the result is therefore printed out as $\sqrt{I_o/I_p} = P_o/P_p$, where P_o is the initial pressure 1 metre from the source and P_p is the pressure on point p.

This amount is equal to the horizontal distance when we have spherical spreading, and, for example, the same 75 dB loss will be printed out as 5624. The value of $\frac{\sqrt{I_o/I_p}}{X}$ is also printed out to show how the loss compares to normal spherical spreading.

Instructions 1 to 19 take care of the initial data, and the main program loop starts at Instr. 20.

Instructions 24 to 34 check if $c_m = c_n$, and if so modify c_n. This is to work in accordance with the ray tracing program.

Instructions 35 to 44 check if $c_n = k$, and if so the sign of the denominator (γ) in the approximated sine term is altered. The computer then stops [Instr. 42] to wait for the next input data, and returns to the start of the main loop. Note that nothing is done with the last input data, and Instr. 42 is only to take care of the sequence in the input data.

A computation of Eq. 28 starts at Instr. 46 and finishes at Instr. 56. The result is added to and stored as result in the C register.

When the intensity is wanted, the last part of the program is called by means of a label.

Instructions 65 to 70 compute Eq. 27 and the final result is printed out at Instr. 74.

An initial value for $\gamma = +1$ has to be put into the D register.

It is not possible to calculate the intensity for a ray starting horizontally, or to ask for the intensity at a point where a ray is horizontal.

For a splitting ray, it is possible to compute the intensity for the ray turning back, but not for the ray bending away.

If one wants to calculate the intensity for a reflected ray, one has to shift the sign on all the r_n' (given as input data) after each reflection. This means that one gives the r_n' as they are printed out from the ray tracing program up to and including the reflection point. Thereafter one gives the r_n' with opposite signs to those printed out until the next reflection, and so on.

2.5 Travel-time program [Fig. 5]

Equation 25 is used. The initial data are k and c_0. The input data are sound speed of each layer the ray is passing, and horizontal distance. The program checks if the sound speed is greater than the k value, and if so, it substitutes the sound speed by the k value.

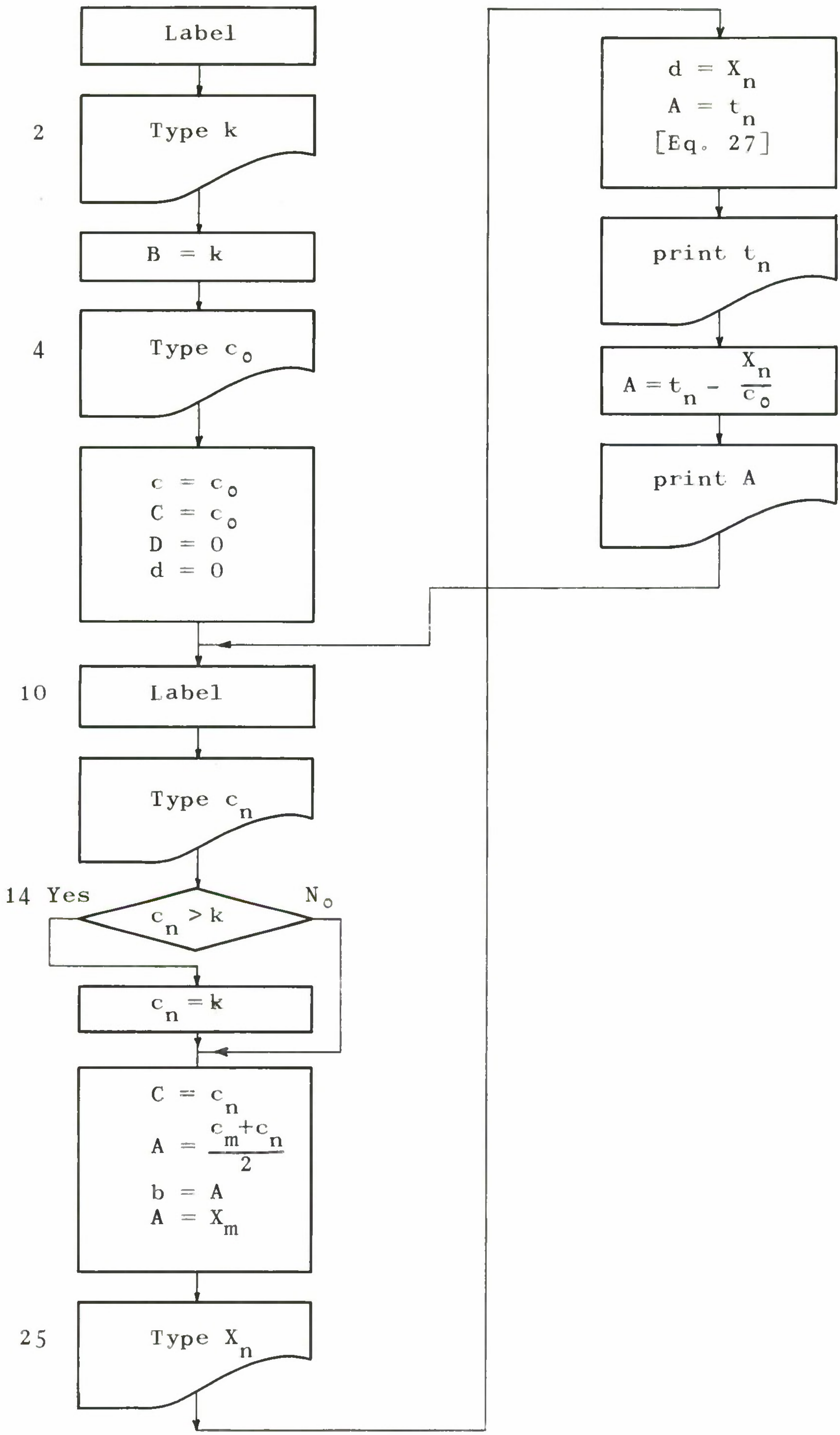

FIG. 5 FLOWCHART – TRAVEL TIME PROGRAM

Output from the program is travel-time t_n and "travel-time anomaly" $\Delta t_n = t_n - X_n/c_0$. The last figure is very convenient when comparing travel-times for different rays.

3. ILLUSTRATION OF THE PROGRAMS

3.1 Complete Ray-tracing Examples

The data are taken from Ref. 7.

Table 2 gives the depth, temperature, and salinity together with the sound speeds computed by means of the sound speed program. As the salinity in this case is more than 30‰ and the depth is less than 7000 m, the use of the "basic formula" is sufficient.

TABLE 2

MEDITERRANEAN, 43°23'N 8°47'E

Date 22 September 1956

Depth	Temperature	Salinity	Computed Sound Speed
0	21.0	38.25	1528.37
10	20.48	38.24	1527.13
20	19.36	38.22	1524.22
30	14.89	38.14	1511.11
50	13.20	38.20	1506.10
75	13.01	38.30	1506.01
90	13.09	38.35	1506.59
160	13.13	38.40	1507.93
210	13.42	38.49	1509.82
310	13.42	38.53	1511.51

Figure 6a shows the sound speed profile.

Suppose now that we have a hull-mounted sonar with a transducer at 10 m depth. (As the source depth then coincides with one of the points of the sound speed profile, the computation of the sound speed for the source depth is superfluous).

The following will explain how to calculate the ray limiting the shadow zone, and some other rays.

By means of the k- value program, one computes the k value for the desired rays, giving sound speed and initial angles at the source as input parameter. As this shape of sound speed profile gives a downward refraction of the rays, the limiting ray will be the one that is horizontal at the sea surface. The k value for a ray is equal to the sound speed at the depth where it is horizontal, hence for this ray the k value is equal to the sound speed at the surface.

The rays from $-3°$ to $+3°$ will be traced. The k value for the rays is the same whether the angle is positive or negative. When the rays going up are traced, one also gets the corresponding rays going down. It is only a matter of shifting the rays a constant horizontal distance until the crossing of the source depth coincides with the source.

Table 3 shows the result of the k value program.

<u>TABLE 3</u>

	V
1527.13	S
1	S
0.99984770	B◊
1527.36261732	A◊
2	S
0.99939083	B◊
1528.06084882	A◊
3	S
0.99862954	B◊
1529.22574271	A◊

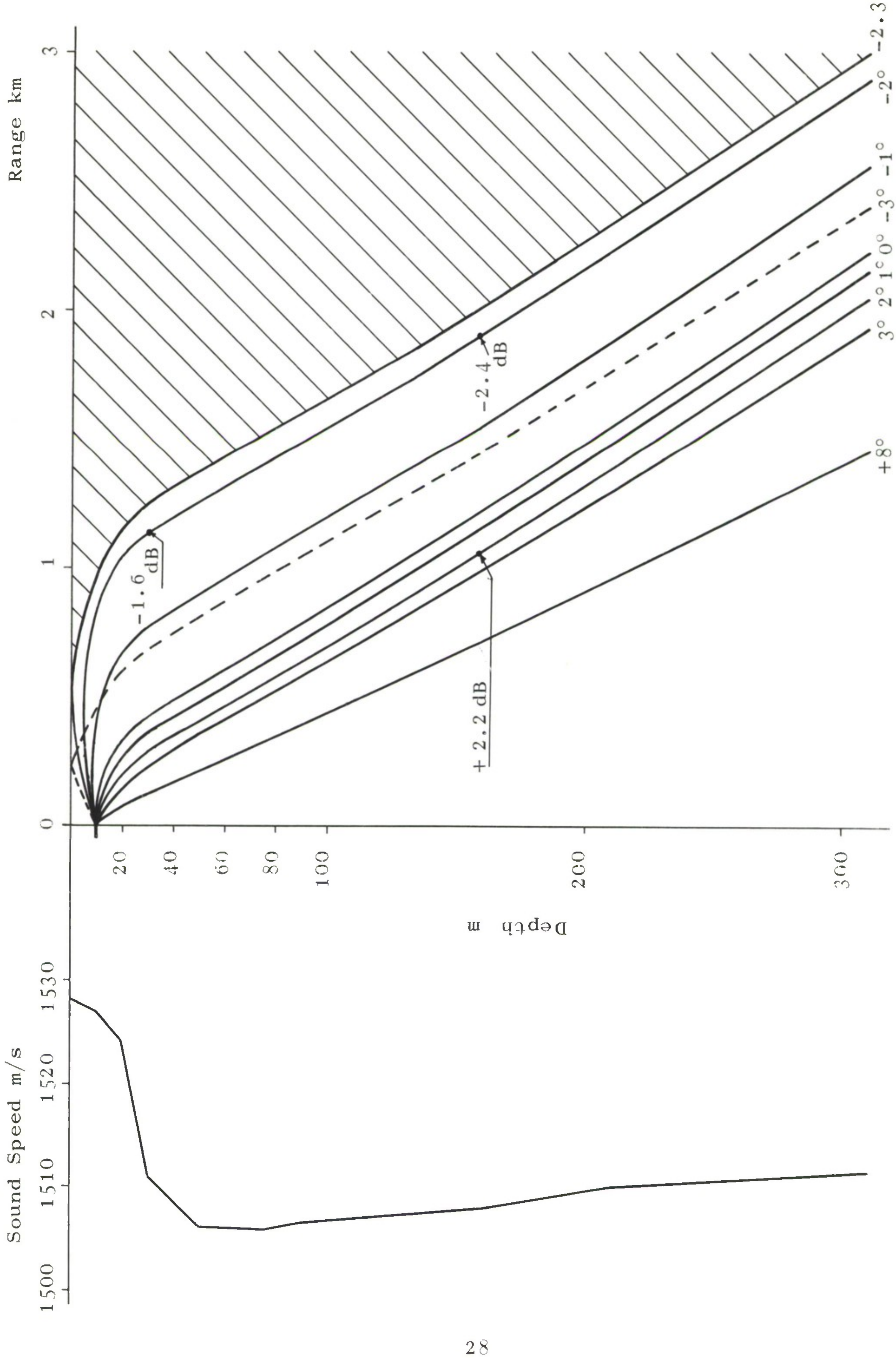

FIG. 6 EXAMPLE OF RAY-TRACING BY DTC

We call the start of the program by V, and give initial data (c_0 = 1527.13 m/s) and press S. Then we give the data ($\theta_0 = 1°$) and press S and the DTC prints cos θ_0(B◊) and the k value (A◊). Thereafter we continue in the same way for other values of θ_0.

The program is used with many decimal places because the cosine for small angles is very near to unity. But we select only two decimal places when we use the result as input data in the ray tracing program.

The computation of the ray starting with a positive angle of 2° is shown in Table 4.

First we call the start of the program by V and give the initial data (k = 1528.06, c_0 = 1527.13 and z_0 = 10), pressing S after each initial data.

The DTC prints out tan θ_0(A◊), which is needed for the intensity program.

Then we ask for horizontal distance to the point where the ray reaches the surface, by giving the sound speed at the surface (1528.37) and the depth as input data. But as the sound speed at the surface is bigger than the k value of the ray, the surface cannot be reached. The computer therefore prints out the depth of the turning point (b◊) and the horizontal distance to the turning point (D◊). The r'(C◊) for the intensity program and the tan θ(A◊) are also printed out.

Next to be asked for is the horizontal distance to the point where the ray returns to the source depth. As the ray will now cross this depth at the same angle as the initial angle, we now also get the data for the ray starting with an initial angle of -2°. The table shows how the computation goes on until the ray is plotted.

TABLE 4

	V
1528.06	S
1527.13	S
10	S
0.03494	A ◊
1528.37	S
0	S
2.5000	b ◊
-445.8233	C ◊
429.9074	D ◊
0.0000	A ◊
1527.13	S
10	S
-445.8233	C ◊
859.8148	D ◊
0.0374	A ◊
1524.22	S
20	S
-189.9728	C ◊
1048.8757	D ◊
0.0722	A ◊
1511.11	S
30	S
-42.1678	C ◊
1139.8527	D ◊
0.1502	A ◊
1506.10	S
50	S
-220.6870	C ◊
1265.4456	D ◊
0.1715	A ◊

1506.01	S
75	S
-15356.1388	C ◊
1412.8645	D ◊
0.1721	A ◊
1506.59	S
90	S
1429.7094	C ◊
1501.7924	D ◊
0.1699	A ◊
1507.93	S
160	S
2887.8708	C ◊
1926.0206	D ◊
0.1642	A ◊
1509.82	S
210	S
1462.4894	C ◊
2241.6258	D ◊
0.1563	A ◊
1511.51	S
310	S
3271.1301	C ◊
2904.6838	D ◊
0.1489	A ◊

The plotting of the other rays is done in the same way, the result being shown in Fig. 6b. It is easily seen from the figure that the ray that is tangent to the surface is the limit of the shadow zone.

As we have no crossing of rays in this case, the travel-time computation is of little interest. The intensity calculation, however, is of great interest, because Fig. 6b clearly shows that the intensity in the domain covered by rays with a positive initial angle is less than that for negative angles.

Table 5 shows intensity calculation for two rays: those with initial angles of $-2°$ and $+2°$. For the ray with $-2°$ initial angle, the intensity is asked for at distances of 1266 m and 1926 m. By taking 20 log of the answer (A◇) one finds the geometric spreading losses to be 66.6 dB and 68.1 dB respectively. By taking 20 log of the next answer, $(A◆) = (\sqrt{I_o/I_p})/X$, one obtains a comparison with the spherical losses. In this case it turns out that the geometrical spreading losses are respectively 1.6 dB and 2.4 dB higher than the spherical spreading losses. For the ray that starts with initial angle $+2°$ the intensity is computed at a distance of 1066 m, and the intensity here is 2.2 dB higher than spherical spreading.

The same data have been run with SACLANTCEN's Ray Tracing Program (C.P. 67) on SACLANTCEN's Elliott 503 computer. The result for one initial angle is shown in Table 6.

Table 7 compares the results of the two ray-tracing programs for the $-2°$ initial angle. Under the travel-time one column shows the correction for the difference in path length.

3.2 Use of a pre-computation to prepare ray-tracing data for a large computer

The above ray-tracing example was to illustrate how to use the DTC for normal ray tracing. Nowadays at most of the places

TABLE 5

-2° initial angle

	V	
1528.06	S	
1527.13	S	
0.03494	S	
1528.37	S	
-445.8	S	
1527.13	S	
-445.8	S	
1524.22	S	
-190	S	
1511.11	S	
-42.2	S	
1506.1	S	
-220.7	S	
	Z	
1265	S	
2149.68090	A	◊
1.69935	A	◊
1506.01	S	
-15356.2	S	
1506.59	S	
1429.7	S	
1507.93	S	
2887.9	S	
	Z	
1926	S	
2544.63568	A	◊
1.32120	A	◊

+2° initial angle

	V	
1528.06	S	
1527.13	S	
0.03494	S	
1524.22	S	
-190	S	
1511.11	S	
-42.2	S	
1506.1	S	
-220.7	S	
1506.01	S	
-15356	S	
1506.59	S	
1429.7	S	
1507.93	S	
2887.9	S	
	Z	
1066	S	
830.18232	A	◊
0.77878	A	◊

TABLE 6

Raytr,A.S.TM

Initial angle -1.999

CODE: 0=D 1=U 2=S 3=B 4=UD 5=DU

CODE	distance m	depth m	travel-time sec	angle degrees	path-length m	intensity dB
4	429.9	2.5	.281422	0.00	429.9	-52.67
0	859.7	10.0	.562846	2.00	859.9	-58.69
0	1048.6	20.0	.686802	4.06	1049.0	-62.13
0	1139.1	30.0	.746811	8.54	1140.1	-65.65
0	1263.5	50.0	.830327	9.73	1266.1	-66.62
0	1409.2	75.0	.928492	9.75	1414.0	-67.07
0	1497.2	90.0	.987712	9.62	1503.2	-67.25
0	1917.1	160.0	1.27018	9.31	1928.9	-68.08
0	2229.8	210.0	1.48002	8.86	2245.6	-68.43
0	3500.0	398.8	2.32967	8.05	3529.7	-69.50

where one does ray tracing, one normally has access to a computer. But even then the DTC and the programs described here can be a help. For a certain sound speed gradient, one can make a short and fast ray tracing by means of the DTC. Afterwards, it is much easier to ask for the correct things from the large computer.

The example given in Fig. 7 will clarify this. The sound speed profile is shown at the left. On the right are some rays plotted by means of the DTC ray-tracing program. The rays selected are those that have turning points at depths where the sound speed gradient has discontinuity points. This simple ray tracing can give a good idea about shadow zones, focusing, multipath area, or caustics, thereby allowing more exact data to be specified when using a large computer.

TABLE 7

COMPARISON BETWEEN THE RESULTS OF RAY-TRACING PROGRAMS WITH A LARGE COMPUTER (ELLIOTT 503) AND WITH A DESK-TOP COMPUTER (DTC)

Depth (m)	Distance (m)		Travel-time (s)			Angle (°)		Intensity (dB)	
	Elliott 503	DTC	Elliott 503	DTC	DTC corr.	Elliott 503	DTC	Elliott 503	DTC
2.5	429.9	429.9	0.2814	0.2814	0.2814	0.00	0.00	-52.67	-
10	859.7	859.8	0.5628	0.5628	0.5627	2.00	2.15	-58.69	-58.69
20	1048.6	1048.9	0.6868	0.6869	0.6867	4.06	4.13	-62.13	-62.14
30	1139.1	1139.9	0.7468	0.7472	0.7468	8.54	8.54	-65.65	-65.66
50	1263.5	1265.4	0.8303	0.8314	0.8301	9.73	9.73	-66.62	-66.64
75	1409.2	1412.9	0.9285	0.9306	0.9282	9.75	9.77	-67.07	-67.09
90	1497.2	1501.8	0.9877	0.9904	0.9874	9.62	9.65	-67.25	-67.28
160	1917.1	1926.0	1.2702	1.2755	1.2698	9.31	9.33	-68.08	-68.11
210	2229.8	2241.6	1.4800	1.4871	1.4794	8.86	8.88	-68.43	-68.46

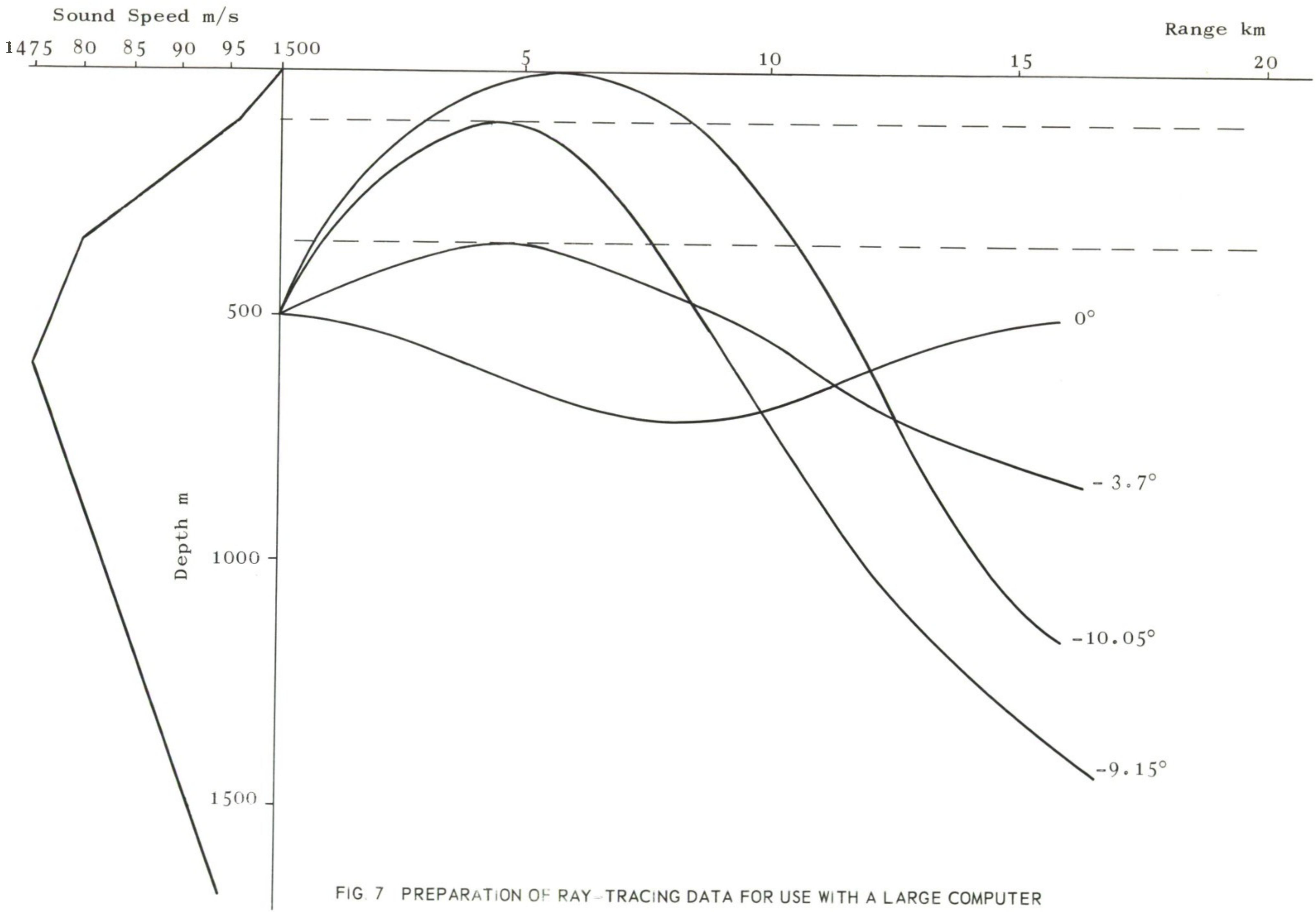

FIG. 7 PREPARATION OF RAY - TRACING DATA FOR USE WITH A LARGE COMPUTER

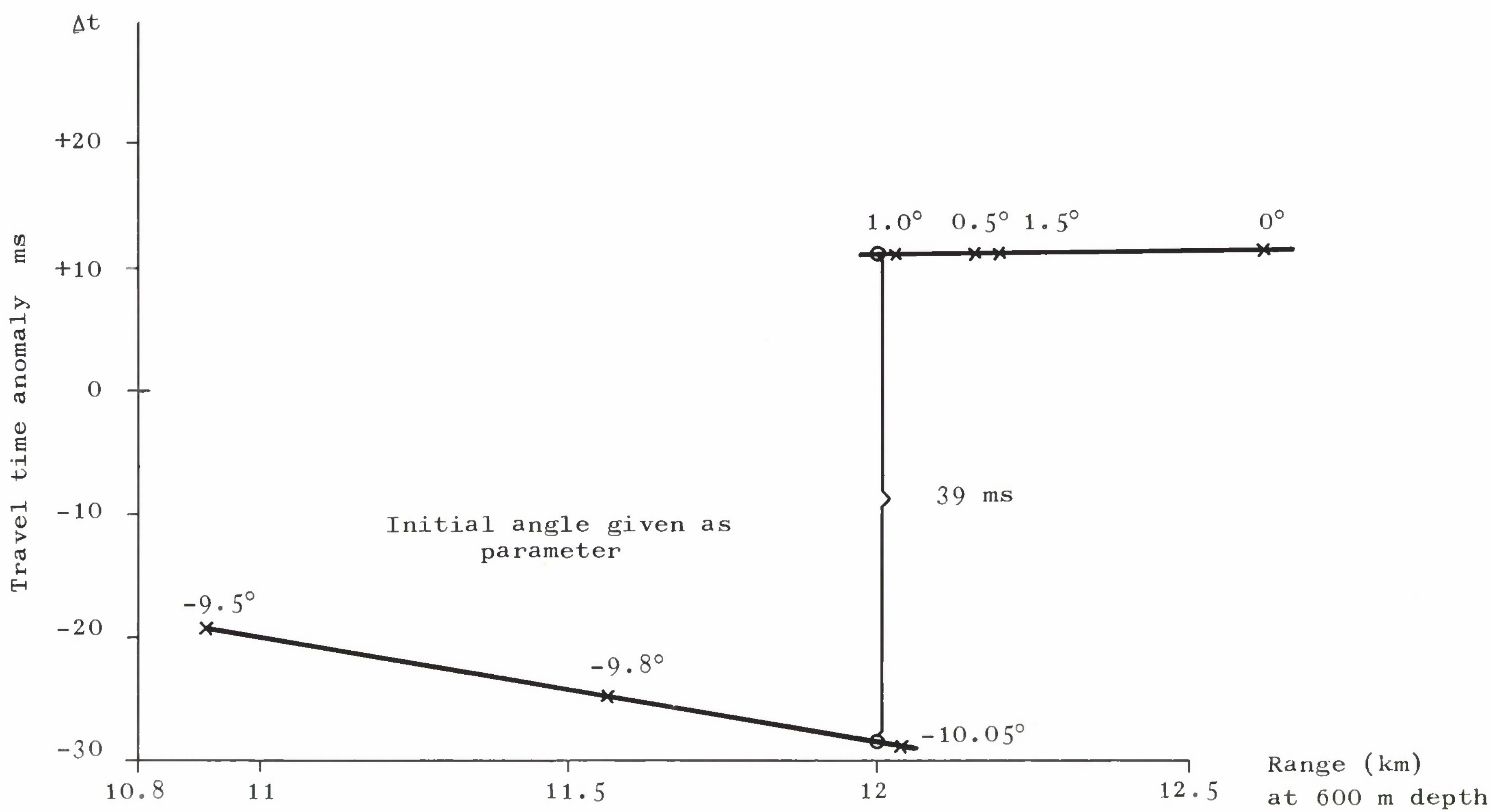

FIG. 8 TRAVEL-TIME ANOMALY AS A FUNCTION OF DISTANCE AT 600 m DEPTH FOR DIFFERENT INITIAL ANGLES

A comparison was made between the results of these different DTC programs and those given by the programs of a large computer, using the same sound speed profile [Fig. 6]. In spite of the comparatively extreme conditions shown by this profile — thick layers, rather large sound speed differences, and initial angles of up to 10° — the comparison revealed that the DTC program makes an error of less than 1% in horizontal range and of less than 1‰ in travel time (corrected for difference in horizontal range). By dividing into thinner layers, the error in corrected travel time can be reduced to less than 0.3‰. The error in intensity program is normally less than 0.1 dB, but in cases with intensity anomalies of the order of 20 dB, the error is less than 1 dB.

3.3 Example of Travel-time Program

From Fig. 7 we see that rays with initial angles of about 0° cross rays with initial angles of about -10°. To analyze this in more detail, some rays have been computed by means of the ray-tracing program and travel-time program. (The layer between 100 m and 350 m was divided into four layers during the computation to get good accuracy.)

In Fig. 8 the travel-time anomaly is plotted as a function of distance at 600 m depth for different initial angles. We notice that when the initial angle increases from 0°, the horizontal distance out to a depth crossing of 600 m first decreases up to about 1° initial angle and afterwards increases. For the other family of rays, we know that for initial angles steeper than -10.05° surface reflection occurs and the horizontal distance decreases sharply. Therefore, it is only at about 12040 m that intersection of the two ray-families can occur at 600 m depth (neglecting short distance with surface-reflected rays).

We see from the figure that the ray with about -10° initial angle arrives about 39 ms before the one with about 1° initial angle.

CONCLUSION

By means of the programs presented here and a DTC, it is possible to compute the sound field in the same manner as with a ray-tracing program for a large computer. Of course the time necessary for the computation is quite different.

For a large program the Olivetti 101 Desk Top Computer uses about 15 seconds for each sequence. It should therefore take about one hour to convert a set of temperature, salinity, and depth profiles into a sound speed profile, make ray-tracing for some rays, and make intensity calculation at some points (assuming about ten layers).

Where a computer is not available, it is thus possible by means of a DTC to make some ray-tracing within a reasonable time. When a computer is available but is heavily booked, the use of these programs on a DTC can be useful to give some results while waiting to obtain the use of the large computer.

PROGRAMS

TABLE 8

TITOLO

PRESSURE-DEPTH PROGRAM

N.	ISTRUZIONI indirizzo	ISTRUZIONI funzione	M	A	B: B/	B: B	C: C/	C: C	D: D/	D: D	E: E/	E: E	F: F/	F: F
1	A	V							k	m	n	p	q	
2		S												
3		↓		z										
4	D/	:		$z/1000$										
5	C	↕						$z/1000$						
6		S												
7		↓		φ										
8	A	↕		φ										
9	/	V												
10		W												
11	A/	V		φ										
12	D	:		x										
13	A	x	x	x^2										
14		x	x	x^3										
15	B	↑				x								
16	A/	↑												
17	E/	x	-6											
18		:		$-x^3/6$										
19	A	x	$-x^3/6$	$x^6/6.6$										
20	B/	↑			$-x^3/6$									
21	B	:		$x^5/6.6$										
22	A/	↑												
23	R	↕												
24	R	↕												

TABLE 8 (Cont'd)

TITOLO
PRESSURE-DEPTH PROGRAM

N.	ISTRUZIONI		M	A	B		C		D		E		F	
	indirizzo	funzione			B/	B	C/	C	D/	D	E/	E	F/	F
25	D/	↕	3.33	$\times\,^{5}/_{36}$	$-\times\,^{3}/_{6}$			$^{z}/_{10^3}$	k	m	n	p	q	
26		:		$\times\,^{5}/_{5}$:										
27	C/	↕					$\times\,^{5}/_{5}$:							
28	C/	↓		$\times\,^{5}/_{5}$:										
29	B/	×		$-\times\,^{8}/_{6}$:										
30	B	:		$-\times\,^{7}/_{6}$:										
31	A/	↑												
32	D/	:	7											
33		:		$-\times\,^{7}/_{7}$:										
34	C/	+												
35	B/	+												
36	B	+												
37	A	W		sin φ										
38	A	×		$\sin^2\varphi$										
39	E	×		0.0528										
40	A/	↑												
41	R/	S	10											
42	D	↓												
43		+		10+0.0528										
44	C	×												
45	E/	×												
46	C	↕		$^{z}/_{10^3}$										
47	A	×		$^{z^2}/_{10^6}$										
48	F/	×												

TABLE 8 (Cont'd)

TITOLO
PRESSURE-DEPTH PROGRAM

N.	ISTRUZIONI		M	A	B		C		D		E		F	
	indirizzo	funzione			B/	B	C/	C	D/	D	E/	E	F/	F
49	C	+												
50	A/	↑												
51	R	+												
52	R	S	1.04											
53	D/	↓												
54		+		p										
55	A	◊												
56	/	◊												
57		V												

TABLE 9

LEROY 2. SOUND SPEED FORMULA

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25	R ◊	49	C ↑	73	B ↕	97	
2	A/↑	26	R/ ↕	50	C/ :	74	B/ ↓	98	
3	R/ S	27	D↓	51	B +	75	A/ ↑	99	
4	D ↓	28	×	52	B ↕	76	R −	100	
5	C/↑	29	B +	53	B/ ↓	77	D/ S	101	
6	S	30	C ↕	54	A/ ↑	78	−	102	
7	↓	31	B/ ×	55	R ◊	79	C/ ↕	103	
8	C/ :	32	C ↕	56	D/ ↓	80	C/ ↓	104	
9	B/ ↕	33	C −	57	−	81	A/ ↑	105	
10	B/↓	34	B ↕	58	C ×	82	D/ ↑	106	
11	A/↑	35	S	59	C/ :	83	−	107	
12	F ×	36	C ↑	60	A ×	84	B/ ×	108	
13	E/ +	37	C ↓	61	A +	85	C/ ×	109	
14	×	38	A/↑	62	B +	86	A/ ↑	110	
15	D +	39	R/ ↓	63	B ↕	87	R ×	111	
16	B/ ×	40	D ×	64	S	88	R ↑	112	
17	D/ +	41	:	65	↓	89	D/ S	113	
18	B ↕	42	B +	66	A/ ↑	90	×	114	
19	S	43	B ↕	67	R/ S	91	B +	115	
20	↓	44	C ↓	68	D *	92	A ◊	116	
21	C/ :	45	C/ :	69	:	93	/ ◊	117	
22	C ↕	46	:	70	C ×	94	V	118	
23	C ↓	47	:	71	C/ :	95		119	
24	A/↑	48	A ×	72	B +	96		120	

CONSTANTS		CONSTANTS	
1401.14	D/ ↑		↑
49.1	D ↑		↑
	↑		↑

TABLE 10

SOUND SPEED - DEEP OCEAN TERM AND LOW SALINITY TERM

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25	B ↑	49		73		97	
2	S	26	C/ ↕	50		74		98	
3	↓	27	D/ ↓	51		75		99	
4	D/ :	28	S	52		76		100	
5	A ×	29	+	53		77		101	
6	B ↑	30	D/ :	54		78		102	
7	B ↕	31	C/ ×	55		79		103	
8	F +	32	F/ ×	56		80		104	
9	B/ ↕	33	D/ :	57		81		105	
10	B/ ↓	34	C ↕	58		82		106	
11	F +	35	B/ ↓	59		83		107	
12	B/ ×	36	E/ −	60		84		108	
13	B ×	37	B ×	61		85		109	
14	D/ :	38	D/ :	62		86		110	
15	A ◊	39	B/ ×	63		87		111	
16	/ ◊	40	×	64		88		112	
17	V	41	D/ :	65		89		113	
18	A W	42	E ×	66		90		114	
19	S	43	C −	67		91		115	
20	B/ ↑	44	A ◊	68		92		116	
21	S	45	/ ◊	69		93		117	
22	↓	46	W	70		94		118	
23	D −	47		71		95		119	
24	A ×	48		72		96		120	

CONSTANTS		CONSTANTS	
-1000	D/ ↑	3	E ↑
35	D ↑	1.5	F/ ↑
30	E/ ↑	4	F ↑

TABLE 11

K-VALUE PROGRAM

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25		49		73		97	
2	S	26		50		74		98	
3	C ↑	27		51		75		99	
4	A W	28		52		76		100	
5	S	29		53		77		101	
6	↓	30		54		78		102	
7	F :	31		55		79		103	
8	A ×	32		56		80		104	
9	F/ :	33		57		81		105	
10	A ×	34		58		82		106	
11	B ↑	35		59		83		107	
12	E :	36		60		84		108	
13	B +	37		61		85		109	
14	E/ +	38		62		86		110	
15	B ↕	39		63		87		111	
16	B ◊	40		64		88		112	
17	C ↓	41		65		89		113	
18	B :	42		66		90		114	
19	A ◊	43		67		91		115	
20	/ ◊	44		68		92		116	
21	W	45		69		93		117	
22	V	46		70		94		118	
23		47		71		95		119	
24		48		72		96		120	

CONSTANTS		CONSTANTS	
1	E/ ↑	57.2958	F ↑
6	E ↑		↑
-2	F/ ↑		↑

TABLE 12

RAY-TRACING PROGRAM

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25	C/ ↕	49	C/ +	73	A ↕	97	
2	S	26	C ↑	50	B −	74	C ◇	98	
3	B ↑	27	A/ V	51	C ↕	75	D +	99	
4	S	28	S	52	B ↓	76	D ↕	100	
5	C/ ↑	29	↓	53	C/ ↕	77	D ◇	101	
6	S	30	B/ ↕	54	B/ ◇	78	A Z	102	
7	B/ ↑	31	B/ −	55	A Y	79	C/ ↓	103	
8	D *	32	D/ ↕	56	B ↓	80	B :	104	
9	Z	33	C/ ↓	57	A +	81	D/ ↕	105	
10	A W	34	B −	58	A √	82	D/ ↓	106	
11	S	35	/ W	59	D/ ×	83	A ×	107	
12	↓	36	Y	60	C :	84	A :	108	
13	C/ ↕	37	A/ W	61	C ↕	85	−	109	
14	C/ −	38	A −	62	A −	86	A √	110	
15	C ↕	39	−	63	−	87	D/ :	111	
16	C ↓	40	D/ ×	64	C/ −	88	A ◇	112	
17	A ↕	41	C :	65	B +	89	/ ◇	113	
18	/ V	42	B/ ↕	66	A √	90	W	114	
19	C/ ↓	43	B/ +	67	D/ ↕	91		115	
20	A/ ↑	44	B/ ↕	68	B ↓	92		116	
21	F ↓	45	D/ ↕	69	C/ −	93		117	
22	F S	46	D/ −	70	A √	94		118	
23	E/ S	47	D/ ↕	71	D/ −	95		119	
24	−	48	C ↓	72	C ×	96		120	

CONSTANTS		CONSTANTS	
	↑		↑
	↑		↑
	↑		↑

TABLE 13

INTENSITY PROGRAM

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25	A ↕	49	↕	73	B/ ×	97	
2	C *	26	/ V	50	D ↓	74	A ◊	98	
3	S	27	D/ ↓	51	:	75	B/ :	99	
4	B ↑	28	A/ ↑	52	E/ ↕	76	:	100	
5	S	29	R ↓	53	E/ −	77	A ◊	101	
6	D/ ↑	30	R S	54	S	78	/ ◊	102	
7	B ↓	31	E/ S	55	×	79	W	103	
8	D/ :	32	+	56	C/ ×	80		104	
9	S	33	D/ ↕	57	C +	81		105	
10	×	34	A/ V	58	C ↕	82		106	
11	C/ ↕	35	B ↓	59	/ ◊	83		107	
12	B ↓	36	D/ −	60	W	84		108	
13	D/ −	37	/ W	61	A Z	85		109	
14	A √	38	D ↓	62	S	86		110	
15	↕	39	A −	63	√	87		111	
16	D ↓	40	−	64	B/ ↕	88		112	
17	:	41	D ↕	65	B ↓	89		113	
18	E/ ↕	42	S	66	A/ ↑	90		114	
19	/ ◊	43	/ ◊	67	D/ ↑	91		115	
20	A W	44	W	68	:	92		116	
21	S	45	A/ W	69	A √	93		117	
22	↓	46	B ↓	70	E/ :	94		118	
23	D/ ↕	47	D/ −	71	C ×	95		119	
24	D/ −	48	A √	72	A √	96		120	

CONSTANTS		CONSTANTS	
1	D ↑		↑
	↑		↑
	↑		↑

TABLE 14

TRAVEL-TIME PROGRAM

REGISTER 1		REGISTER 2		REGISTER F		REGISTER E		REGISTER D	
1	A V	25	S	49	W	73		97	
2	S	26	−	50		74		98	
3	B ↑	27	D/ ↑	51		75		99	
4	S	28	A ↕	52		76		100	
5	C/ ↑	29	E ↕	53		77		101	
6	C ↑	30	B ↓	54		78		102	
7	D *	31	B/ −	55		79		103	
8	D/ *	32	E ×	56		80		104	
9	/ ◊	33	E/ ×	57		81		105	
10	A W	34	B :	58		82		106	
11	S	35	:	59		83		107	
12	↓	36	D +	60		84		108	
13	B −	37	D ↕	61		85		109	
14	/ V	38	D/ ↓	62		86		110	
15	B +	39	B :	63		87		111	
16	Y	40	D +	64		88		112	
17	A/ V	41	A ◊	65		89		113	
18	B ↓	42	B/ ↕	66		90		114	
19	A Y	43	D/ ↓	67		91		115	
20	C ↕	44	C/ :	68		92		116	
21	C +	45	B/ ↕	69		93		117	
22	E/ :	46	B/ −	70		94		118	
23	B/ ↕	47	A ◊	71		95		119	
24	D/ ↓	48	/ ◊	72		96		120	

CONSTANTS		CONSTANTS	
2	E/ ↑		↑
	↑		↑
	↑		↑

APPENDIX A

APPLICATION OF PROGRAMS

TO AN OLIVETTI 101 DESK-TOP COMPUTER

APPENDIX A

The appendix contains first a brief description of an Olivetti 101 Desk Top Computer, enabling the reader to follow the programming.

Then follows an easy program (pressure-depth program) in which every instruction in the programming is explained.

The last part contains some remarks about the application of the formulae to an Olivetti 101 together with the complete programs in Tables. Instructions for the use of an Olivetti 101 are also given.

A.1 THE OLIVETTI 101 ELECTRONIC DESK-TOP COMPUTER

The desk-top computer works with programs recorded on magnetic cards. As shown in Fig. A.1 (taken from Ref. 8) it consists of the following basic units:

a. A Memory to hold numerical data and program instructions.

b. A Keyboard to put data or instructions into the computer.

c. A Printing Unit.

d. A Read or Recording Unit for the magnetic cards.

e. A Control and Arithmetic Unit to receive instructions from the memory, interpret their meeting, and carry them out through the arithmetic unit.

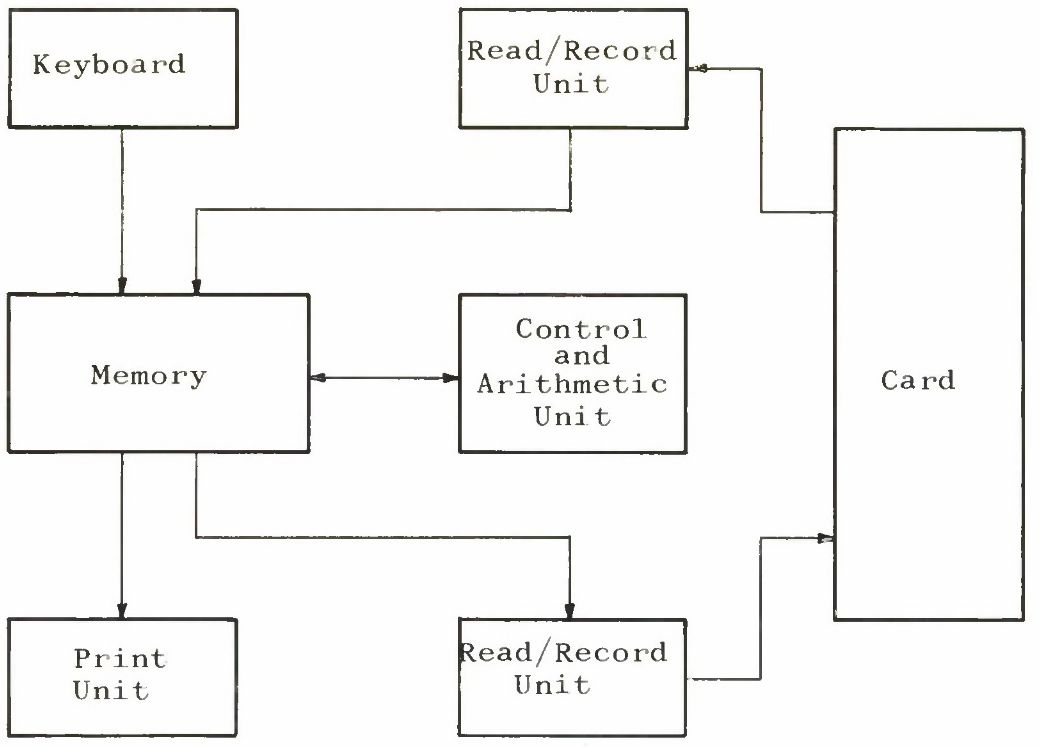

FIG. A.1 BLOCK DIAGRAM OF THE OLIVETTI 101 ELECTRONIC DESK-TOP COMPUTER

The memory consists of ten registers — two program instruction registers, three operational registers, and five storage registers. Each of the storage registers can be split into two parts. The two program instruction registers can store 48 instructions. If more than this number are given, the computer starts to fill one of the five storage registers with instructions and it can then no longer be used for storage. Up to three of the storage registers can be filled with instructions, thus enabling the computer to take 120 instructions.

The computer is capable of normal arithmetical operations and in addition can:

a. Give absolute values.

b. Extract square root of absolute values.

c. Generate constants by means of pseudo-program instructions.

d. Make conditional and unconditional jumps in the instruction sequence.

The content of the two program instruction registers and three of the storage registers can be recorded on a magnetic card, and afterwards read into the computer by the same card. This means that the "program card" can contain only program instructions, or can contain program instructions and constants.

A.2 PROGRAM FOR THE USE OF A PRESSURE FORMULA

A simple formula [Ref. 9], which gives the absolute pressure in the ocean as a function of depth and latitude, has been proposed by C.C. Leroy. This formula has sufficient accuracy for use in underwater acoustics.

The formula is:

$$P(z) = 1.04 + 0.102506\ (1 + 0.00528\ \sin^2 \varphi)\ z + 2.524 \times 10^{-7}\ z^2,$$

where P is in kg/cm², z is in metres, and φ is the latitude in degrees.

To compute this formula on a DTC, it is first necessary to make a program for sin φ. This is done by computing sin φ after the mathematical series

$$\sin \varphi = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} \ldots\ldots,$$

X is the corresponding value in radians.

In the program, four terms of the series are used. The pressure formula may be written as follows:

$$P(z) = 1.04 + 10.2506(10 + 0.0528\ \sin^2 \varphi)\ z/10^3 + 0.2524\ z^2/10^6.$$

The accuracy of the sin φ part of the program is best for small angles, but even for 90° the result is 0.99992.

The program is shown in Table 8 of the main text and will be explained in detail.

The program consists of 57 instructions, which means that two program registers and half the F register are occupied by program instructions. This leaves space for five constants. These constants are put into the memory after the program instructions are made, and afterwards both the program instructions and the constants are recorded on the magnetic card.

For convenience, the constants used are each allocated a letter:

$k = 1000$, $m = 57.29578$, $n = 10.2506$, $p = 0.0528$, $q = 0.2524$.

It is important to note that the arithmetical operations take place with the A and M register, so the content of a register is first transferred into the M register before an arithmetical operation takes place together with the A register.

An explanation of the various instructions follows:

1. Label at beginning of the program. N.B. the constants are automatic in their registers, when the program starts.

2. Stop to receive input from keyboard.

3. Put input data (z = depth) into A register.

4. Divide content of A register by content of d register.

5. Exchange content of A register with content of C register (which means here: put content of A register into C register).

6. Stop to receive input from keyboard.

7. Put input data (latitude in degrees) into A register.

8. Take absolute value of content of A register.

9. Conditional jump. If content of A register is negative or zero, the computer goes on in the sequence; if it is positive the computer jumps to A/V. As the absolute value of the content in the A register is taken, it goes on in the sequence only if the input latitude is zero.

10. Means go to label AW.

11. See 9.

12. Divide content of A by content of D; convert degrees to radians.

13. Multiply content of A by itself and therefore also transfer the content of A (before the multiplication) into M.

14. Multiply A and M, and now we have X^3 in the A.

15. Transfer content of M into B.

16. and 17. Generate the figure -6 in M.

18. Divide A by M, and we now have $-X^3/6 = -X^3/3!$ in A.

19. Multiply content of A register by itself, also transferring content of A (before multiplication) into M.

20. Transfer content of M $(-X^3/6)$ into b.

21. Divide content of $A(X^6/36)$ by content of B.

22, 23, 24 and 25. Generate the figure 3.33 in M.

26. Divide content of A by content of M, making content of A equal to $X^5/5!$.

27. Exchange content of A with content of c.

28. Transfer content of c into A (also leaving content of c unchanged).

29. Multiply content of A by content of b.

30. Divide content of A by content of B.

31, and 32. Generate figure 7 in M.

33. Divide content of A by content of M. Now the content of A is $-X^7/7!$.

34. Add content of c ($X^5/5!$) to content of A.

35. Add content of b ($-X^3/3!$) to content of A.

36. Add content of B(X) to content of A. Now the content of A equal $X - X^3/3! + X^5/5! - X^7/7!$.

37. Label which the computer jumps to in case the input data equal $0°$ (see 10).

38. Multiply content of A by itself, making $\sin^2 \varphi$ in A.

39. Multiply contents of E with contents of A, making $0.0528 \sin^2 \varphi$ in A.

40, 41, 42. Generate figure 10 in M.

43. Add content of M to content of A making $10 + 0.0528 \sin^2 \varphi$ in A.

44. Multiply content of C by content of A, making $(10 + 0.0528 \sin^2 \varphi) Z/1000$ in A.

45. Multiply content of e by content of A, making $10.2506(10 + 0.0528 \sin^2 \varphi) Z/1000$ in A.

46. Exchange content of C with contents of A, which means that the result in A is put into C, and the content of C ($Z/1000$) is put into A.

47. Multiply the content of A with itself making $Z^2/10^6$ in A.

48. Multiply content of f with content of A, making $0.2524\ Z^2/10^6$.

49. Add content of A to content of C, making $10.2506\ (10 + 0.0528 \sin^2 \varphi) Z/1000 + 0.2524\ Z^2/10^6$ in A.

50, 51, 52, 53. Generate the figure 1.04 in M.

54. Add the content of M to the content of A, completing the formula in A.

55. Print out result of A.

56. Advance the paper without printing.

57. Instruction to go to label AV (at beginning of the program).

Constants to be put into register:

1000	into register d
57.29578	into register D
10.2506	into register e
0.0528	into register E
0.2524	into register f

Instruction for use of the program

a. Read the program into the DTC from the magnetic card (the knob Reg. Progr. must be at its outermost position).

b. Put the decimal selector on 5. This is because of the accuracy during the computation, and not because we want there to be this number of decimal places in the result.

c. Call program by pressing V.

d. Print z (depth in metres) on keyboard and press S.

e. Print φ (latitude in degrees) on keyboard and press S.

f. Read result (pressure in kg/cm^2) printed out.

g. Give new data.

A.3 SOUND SPEED PROGRAMS

The programs for the main formula contain a lot of instructions, so there are only two half registers for constants: 1401.14 in d and 49.1 in D. The program is shown in Table 9 and the flowchart in Fig. 2 of the main text.

The program for Vc and Vd has less than 48 instructions, so there is space for six constants in the registers.

These are

-1000	d
35	D
30	e
3	E
1.5	f
4	F

The program is shown in Table 10 of the main text.

Instruction for Use

Main Sound Speed Program

Read the program into the DTC. Put decimal selector on 4 and call program by V.

Input data:

t : temperature in degrees centigrade.
s : salinity in ‰.
z : depth in metres.
φ : latitude in degrees.
Read sound speed in m/s
Give new data.

Low salinity or deep ocean terms

Put decimal selector on 3.

If Vc (deep ocean) is wanted, press V and give depth in m as input data, read Vc in m/s. Vc must then be added to the sound speed calculated by means of the program for the basic formula. If Vd (low salinity term) is wanted, press W. Give temperature (°C), salinity (‰), depth (m) as input, read Vc (m/s) as output. Vc must then be added to the basic formula.

A.4 PROGRAM TO COMPUTE k-VALUE

The program is shown in Table 11 of the main text.

Read the program into the DTC, put decimal selector on 8 and call program by V.

Initial data:

c_o: sound speed at source in m/s.

Data:

θ_o: initial angle in degrees.

Result:

$\cos\theta_o$ and $k = \dfrac{c_o}{\cos\theta_o}$.

New data:

For new initial data, press V.

A.5 RAY TRACING PROGRAM

The flowchart for the program is shown in Fig. 3 and the program in Table 12 of the main text.

Read the program into the DTC.

Decimal selector: during initial data computation on 5, otherwise on 4.

Call program by V.

Initial data:

k : ray constant $= \dfrac{c_n}{\cos\theta_n}$,

c_o : sound speed at source m/s ,

z_o : depth of source in m .

Read $\tan\theta_o$, (θ_o initial angle). This result is needed for the intensity program.

Data:

c_n: sound speed in m/s in layer n (n = 1, 2........),

z_n: depth of layer n.

Read

r'_n needed for intensity program,

X_n horizontal distance in m,

tan θ_n, where θ_n is angle out of layer n.

Data for next layer.

If a sound speed is given that is bigger than the k value of the ray, a turning point is passed. The DTC then prints out the depth of the turning point and modifies the input data.

After a turning point or a reflection, one has to trace the ray back, giving once more the depths and their corresponding sound speeds as data.

If the operator does not notice that a turning point is passed and asks for a new point with a sound speed higher than the k value of the ray, then the DTC stops the computation and the red light comes on. The main loop of the program can be called by W, and the data stored in the computer are still correct.

When a ray is traced, press V and give initial data for next ray.

A.6 INTENSITY PROGRAM

The flowchart of the program is given in Fig. 4 and the

program in Table 13 of the main text. Constants to be put in: 1 into D.

The program is based on the ray tracing program being used first, and the values for k, tan θ_0, and r_n' are therefore known.

Load the program by means of the magnetic card, and put decimal selector on 5.

Call program by means of V.

Give initial data

- k constant for the ray,
- c_0 sound speed at the source,
- tan θ_0, where θ_0 is initial angle.

Thereafter give

- c_n sound speed in layer n (n = 1, 2.......),
- r_n' modified curvature radius printed out from ray tracing program (the sign is important) until layer p, where one wants the intensity, is reached, and until c_p and r_p' are given as input data.

Then press Z and give X_p (horizontal distance to the point where the ray leaves the layer p).

The outputs are $\sqrt{I_0/I_p}$ and $\dfrac{\sqrt{I_0/I_p}}{X}$.

If more intensity points are wanted, continue by giving c_{p+1} and r'_{p+1} as input data, and so on.

For limitation of the program, see Sect. 2.4 of main text.

A.7 TRAVEL TIME PROGRAM

The flowchart for the program is shown in Fig. 5. and the program in Table 14 of the main text. Constants to be put into register: 2 into e.

Read the program into the DTC.

Decimal selector on 4, call program by V.
Initial data:

k : constant for the ray,

c_0 : sound speed at source (m/s).

Data:

c_n : sound speed in m/s in layer n, ($n = 1, 2, \ldots\ldots$),

X_n : corresponding horizontal distance (m) given by ray tracing program.

Result:

t : travel time in seconds,

$t - \frac{X_n}{c_0}$: travel time anomaly.

For new initial data, press V.

If c_n is given bigger than k, this means that a turning point is reached and the computer substitutes c_n by k.

REFERENCES

1. "Physics of Sound in the Sea", NDRC Summary Technical Reports.

2. G.F. Graber, "Report on Acoustic Ray Tracing on the General Purpose Electronic Analog Computer", Princeton Computation Center 1960.

3. D.M. Furuike, "Computerized Calculation and Plotting of Significant Sound Ray Paths and Propagation Loss for Single Temperature Depth (BT) Profiles", T.M. No. 10, 1965, Fleet Numerical Weather Facility, Monterey, California.

4. C.C. Leroy, "Development of Simple Equations for Accurate and more Realistic Calculation of the Speed of Sound in Sea Water", SACLANTCEN Technical Report No. 128, 15 November 1968, NATO UNCLASSIFIED.

5. J.W. Horten, "Fundamentals of Sonar", United States Naval Institute, Annapolis, Maryland, 1957.

6. C.B. Officer, "Introduction to the Theory of Sound Transmission", McGraw-Hill Book Company Inc., 1958.

7. Conseil Permanent International pour l'exploration de la mer, "Bulletin Hydrographique pour l'année 1956", Andr. Fred. Host et Fils, Copenhague.

8. Programming Manual for "Olivetti Programma 101, Electronic Desk-top Computer", Olivetti.

9. C.C. Leroy, "A Universal Formula for the Calculation of Underwater Pressure in Acoustic Studies", SACLANTCEN Technical Report No. 108, 15 March 1968, NATO UNCLASSIFIED

DISTRIBUTION

MINISTRIES OF DEFENCE

	Copies
MOD Belgium	5
MOD Canada	10
MOD Denmark	10
MOD France	8
MOD Germany	13
MOD Greece	11
MOD Italy	10
MOD Netherlands	10
MOD Norway	10
MOD Portugal	5
MOD Turkey	3
MOD U.K.	20
SECDEF U.S.	71

NATO AUTHORITIES

SECGEN NATO	3
NATO Military Committee	2
SACLANT	3
SACEUR	3
CINCHAN	1
SACLANTREPEUR	1
COMNAVSOUTH	1
CINCWESTLANT	1
COMSUBEASTLANT	1
COMCANLANT	1
COMOCEANLANT	1
COMEDCENT	1
COMSUBACLANT	1
COMSUBMED	1
CDR Task Force 442 (via COMSTRIKFORSOUTH)	1
COMMARAIRMED	1

SCNR FOR SACLANTCEN

SCNR Belgium	1
SCNR Canada	1
SCNR Denmark	1
SCNR Germany	1
SCNR Greece	1
SCNR Italy	1
SCNR Netherlands	1
SCNR Norway	1
SCNR Turkey	1
SCNR U.K.	1
SCNR U.S.	1

NATIONAL LIAISON OFFICERS

NLO Italy	1
NLO Portugal	1
NLO U.K.	1
NLO U.S.	1

NLR TO SACLANT

NLR Belgium	1
NLR Canada	1
NLR Denmark	1
NLR Germany	1
NLR Greece	1
NLR Italy	1
NLR Norway	1
NLR Portugal	1
NLR U.K.	1